**TASK 1**

**Conceptual Design through FTR**

**Aim:**

Using basic database design methodology and ER modeler, design Entity Relationship Diagram by satisfying the following sub tasks:

1. a Identifying the entities.
2. b Identifying the attributes.
3. c Identification of relationships, cardinality, type of relationship.
4. d Reframing the relations with keys and constraint.
5. e Using create, develop ER/ER diagram.

**1.a Identifying the Entities**

1.a.1 University  
1.a.2 Department  
1.a.3 Lecturer  
1.a.4 Student  
1.a.5 Course  
1.a.6 Exam

**1.b Identifying the Attributes**

1.b.1 University(UniversityID, Name, Address, Contact\_No)  
1.b.2 Department(DeptID, Name, Head, Location)  
1.b.3 Lecturer(LecturerID, FName, LName, Age, DateofBirth, Qualification, email, contact\_no)  
1.b.4 Student(StudentID, FName, LName, Age, DateofBirth, email, contact\_no)  
1.b.5 Course(CourseID, Name, Credits, Duration)  
1.b.6 Exam(ExamID, Date, Time, Result)

**1.c Identification of Relationships, Cardinality, Type of Relationship**

1.c.1 University–Department Relationship:  
A University will have a **one-to-many** relationship with Departments since a University can have multiple departments, but a department belongs to only one university.

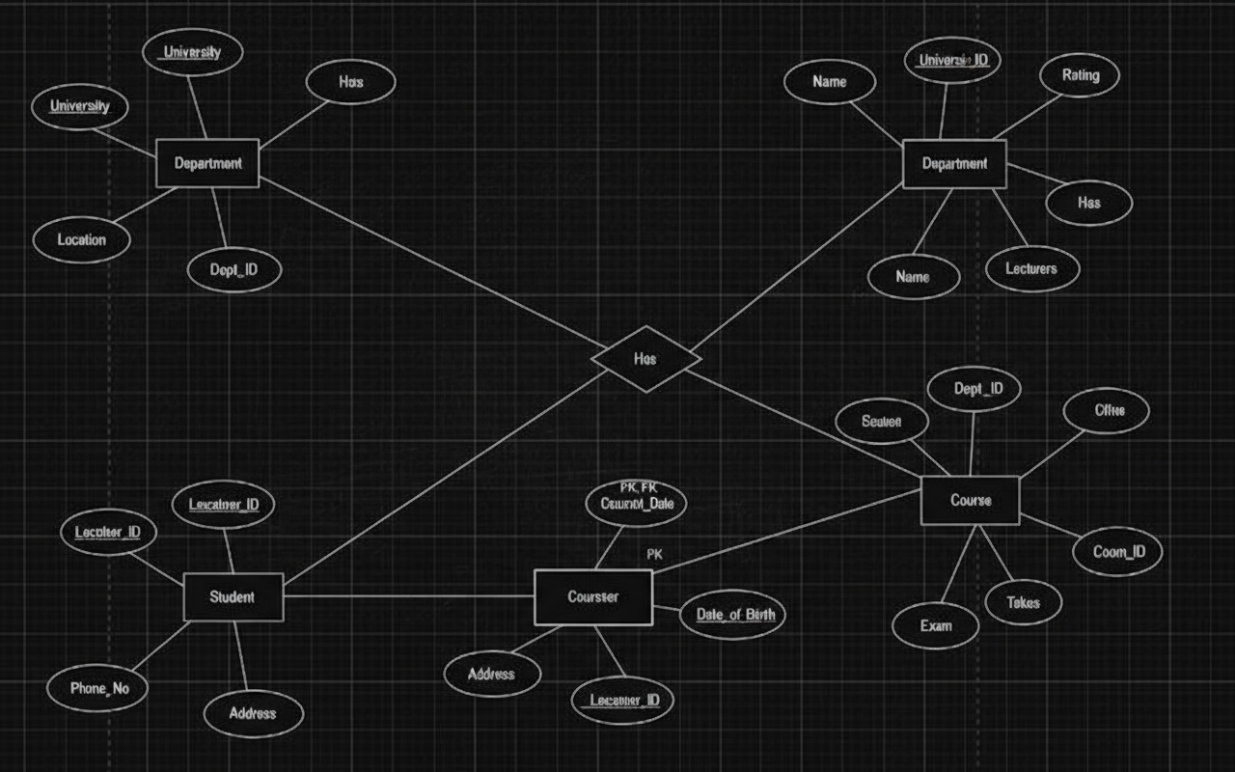
1.c.2 Department–Lecturer Relationship:  
Departments and Lecturers will have a **one-to-many** relationship since one department can have multiple lecturers, but a lecturer belongs to one department.

1.c.3 Department–Student Relationship:  
Departments and Students will have a **one-to-many** relationship since one department can have multiple students, but a student belongs to one department.

1.c.4 Course–Student Relationship:  
Courses and Students will have a **many-to-many** relationship since a student can enroll in multiple courses, and a course can have multiple students.

1.c.5 Exam–Course Relationship:  
Exams will have a **one-to-one** relationship with Courses, as each course has one exam schedule.

**ER Diagram**



**1.d Reframing the Relations with Keys and Constraints**

**1.d.1 Create Table University**

SQL> create table University(

UniversityID varchar(10) PRIMARY KEY,

Name varchar(50),

Address varchar(100),

Contact\_No number

);

Table Created.

SQL> DESC University

Column NULL TYPE

-------------- -------- -----------------

UniversityID NOT NULL VARCHAR2(10)

Name VARCHAR2(50)

Address VARCHAR2(100)

Contact\_No NUMBER

**1.d.2 Create Table Department**

SQL> create table Department(

DeptID varchar(10) PRIMARY KEY,

UniversityID varchar(10),

Name varchar(50),

Head varchar(50),

Location varchar(50),

FOREIGN KEY(UniversityID) REFERENCES University(UniversityID)

);

Table Created.

SQL> DESC Department

Name Null? Type

------------ --------- -----------------

DEPTID NOT NULL VARCHAR2(10)

UNIVERSITYID NOT NULL VARCHAR2(10)

NAME VARCHAR2(50)

HEAD VARCHAR2(50)

LOCATION VARCHAR2(50)

**1.d.3 Create Table Lecturer**

SQL> create table Lecturer(

LecturerID varchar(10) PRIMARY KEY,

DeptID varchar(10),

FName varchar(30),

LName varchar(30),

Age number(5,2),

DateofBirth date,

Qualification varchar(40),

email varchar(50),

contact\_no number,

FOREIGN KEY(DeptID) REFERENCES Department(DeptID)

);

Table Created.

SQL> DESC Lecturer

Name Null? Type

------------ --------- -----------------

LECTURERID NOT NULL VARCHAR2(10)

DEPTID NOT NULL VARCHAR2(10)

FNAME VARCHAR2(30)

LNAME VARCHAR2(30)

AGE NUMBER(5,2)

DATEOFBIRTH DATE

QUALIFICATION VARCHAR2(40)

EMAIL VARCHAR2(50)

CONTACT\_NO NUMBER

**1.d.4 Create Table Student**

SQL> create table Student(

StudentID varchar(10) PRIMARY KEY,

DeptID varchar(10),

FName varchar(30),

LName varchar(30),

Age number(5,2),

DateofBirth date,

email varchar(50),

contact\_no number,

FOREIGN KEY(DeptID) REFERENCES Department(DeptID)

);

Table Created.

SQL> DESC Student

Name Null? Type

------------ --------- -----------------

STUDENTID NOT NULL VARCHAR2(10)

DEPTID NOT NULL VARCHAR2(10)

FNAME VARCHAR2(30)

LNAME VARCHAR2(30)

AGE NUMBER(5,2)

DATEOFBIRTH DATE

EMAIL VARCHAR2(50)

CONTACT\_NO NUMBER

**1.d.5 Create Table Course**

SQL> create table Course(

CourseID varchar(10) PRIMARY KEY,

DeptID varchar(10),

Name varchar(50),

Credits number,

Duration varchar(20),

FOREIGN KEY(DeptID) REFERENCES Department(DeptID)

);

Table Created.

SQL> DESC Course

Name Null? Type

------------ --------- -----------------

COURSEID NOT NULL VARCHAR2(10)

DEPTID NOT NULL VARCHAR2(10)

NAME VARCHAR2(50)

CREDITS NUMBER

DURATION VARCHAR2(20)

**1.d.6 Create Table Exam**

SQL> create table Exam(

ExamID varchar(10) PRIMARY KEY,

CourseID varchar(10),

Exam\_Date date,

Time1 number,

Result varchar(20),

FOREIGN KEY(CourseID) REFERENCES Course(CourseID)

);

Table Created.

SQL> DESC Exam

Name Null? Type

EXAMID NOT NULL VARCHAR2(10)

COURSEID NOT NULL VARCHAR2(10)

EXAM\_DATE DATE

TIME1 NUMBER

RESULT VARCHAR2(20)

**1.d.7 Create Table Student\_Course**

SQL> create table Student\_Course(

StudentID varchar(10),

CourseID varchar(10),

FOREIGN KEY(StudentID) REFERENCES Student(StudentID),

FOREIGN KEY(CourseID) REFERENCES Course(CourseID)

);

Table Created.

SQL> DESC Student\_Course

Name Null? Type

STUDENTID NOT NULL VARCHAR2(10)

COURSEID NOT NULL VARCHAR2(10)

**Result:**

Thus, the database design methodology and ER Model design diagram for the **University** system has been completed successfully.